

SISR-Fluorescent Fiji/ImageJ Plugin: User Manual

Ver. 1.0.0, Oct 30, 2018

Software setup

This plugin is developed with Fiji/ImageJ version 1.5.2e. You can check it by clicking Help->About ImageJ.

The latest version of this software is available at: <https://github.com/whd0121/ImageJ-SISR>

1. Download all the files at <https://github.com/whd0121/ImageJ-SISR> (Click on the Green **Clone or download** button on top-left of the web page and select Download ZIP.)

Unzip the downloaded file. In the rest of this document, we refer to the location of this unzipped folder (contains `Fiji.app(for_SISR)`) as: `[FIJI_PATH]`.

2. Download the `SISR_Plugin` package which is available online as a *Supplementary Software* of our manuscript.
 - Unzip the downloaded files.
 - Move the `SISR_Plugin/DLFL` folder to `[FIJI_PATH]/Fiji.app(for_SISR)/`. You should also see an `ImageJ-win64.exe` in the destination folder.
 - Move all the other files (7 .jar files) to `[FIJI_PATH]/Fiji.app(for_SISR)/plugins/` and replace the existing files.
3. Download the desired pre-trained models (e.g. `Wide_field_TxRed`) which are available online as *Supplementary Software* of our manuscript.
 - Unzip the downloaded files.
 - Move the model folder (e.g. `Wide_field_TxRed`) to `[FIJI_PATH]/Data/Models/`. You should see a `Copy_models_here.txt` file in the destination folder.
4. [Caution] There might be 2 levels of folders after unzipping each file, e.g. `Wide_field_TxRed/Wide_field_TxRed/...` Please move the **inner** folder to the destination folder, e.g., move the `Wide_field_TxRed` folder which contains a `variables` folder and a `saved_model.pb` file to `[FIJI_PATH]/Data/Models/`. After this, the destination folder should look like: `[FIJI_PATH]/Data/Models/Wide_field_TxRed/saved_model.pb`

After downloading our files and moving them to the right locations, please:

5. Double-click on the application: **ImageJ-win64.exe** file in `[FIJI_PATH]/Fiji.app(for_SISR)` folder.
6. An automated Updater option might appear. Please do **NOT** update as new ImageJ core has API bugs.

7. On the application, go to **Process** Menu and select **SISR Transform Network** at the bottom, and start testing various image data included in this package.

Preparing test data

The Plugin works with 8-bit/16-bit grayscale TIF images of size 1024*1024.

Example images are provided in [\[FIJI_PATH\]/Data/Example_images](#).

Please use the correct pre-trained model with the corresponding test images.

Testing a pre-trained model

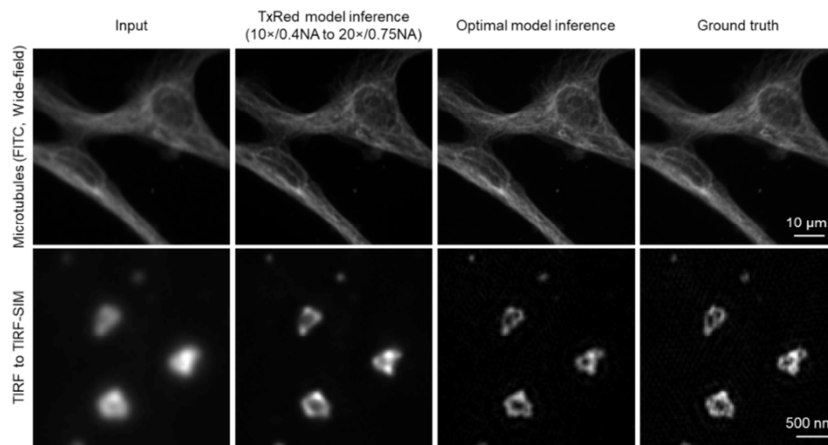
We emphasize that a pre-trained model generates optimal results for input images acquired with the same image formation parameters, including e.g., the magnification, numerical aperture, etc. For users applying our pre-trained models to their own datasets, we recommend starting with TxRed model, although we emphasize that a “new” network model should be trained for achieving optimal super-resolution performance on input images corresponding to different types of samples or captured with a new experimental setup or microscope.

1. In ImageJ, Go to **Process->SISR Transform Network** at the bottom.
2. Select '**Test Single**' option or '**Test Folder**' option:
 - Test Single: open and enhance a single image
 - Test Folder: run the model on all images inside a folder
3. Select Model directory by clicking 'Browse'. For example, select [\[FIJI_PATH \]/Data/Models/Wide_field_TxRed](#) or other models. All the folders are within the same unzipped folder. Once the Model folder is selected, click OPEN. *Please select the model folder, instead of selecting any files inside.*
4. If the **Test Single** Option is selected:
 - Select a single image to be used as Input image to the Model. For example: [\[FIJI_PATH \]/Data/Example_images /Wide_field/TxRed/input/\[imagename\].tif](#)
Depending on the Model you have selected, please use the Input images from the corresponding Input folder. For example, if your model is: [\[FIJI_PATH \]/Data/Models/Wide_field TxRed](#) then select an image from [\[FIJI_PATH\]/Data/Example_images /Wide_field/TxRed/input](#) folder. Same idea applies for all our models.
 - Click Open.
 - The input image will be shown in a separate window.
 - The network output will show up in a few seconds (depending on your machine).
5. If the **Test Folder** Option is selected:
 - Select an input image directory. For example:
[\[FIJI_PATH\]/Data/Example_images/Wide_field/TxRed/input](#)

- Select an output image directory name.
- Plugin will create a folder with network inferred images using the selected model.

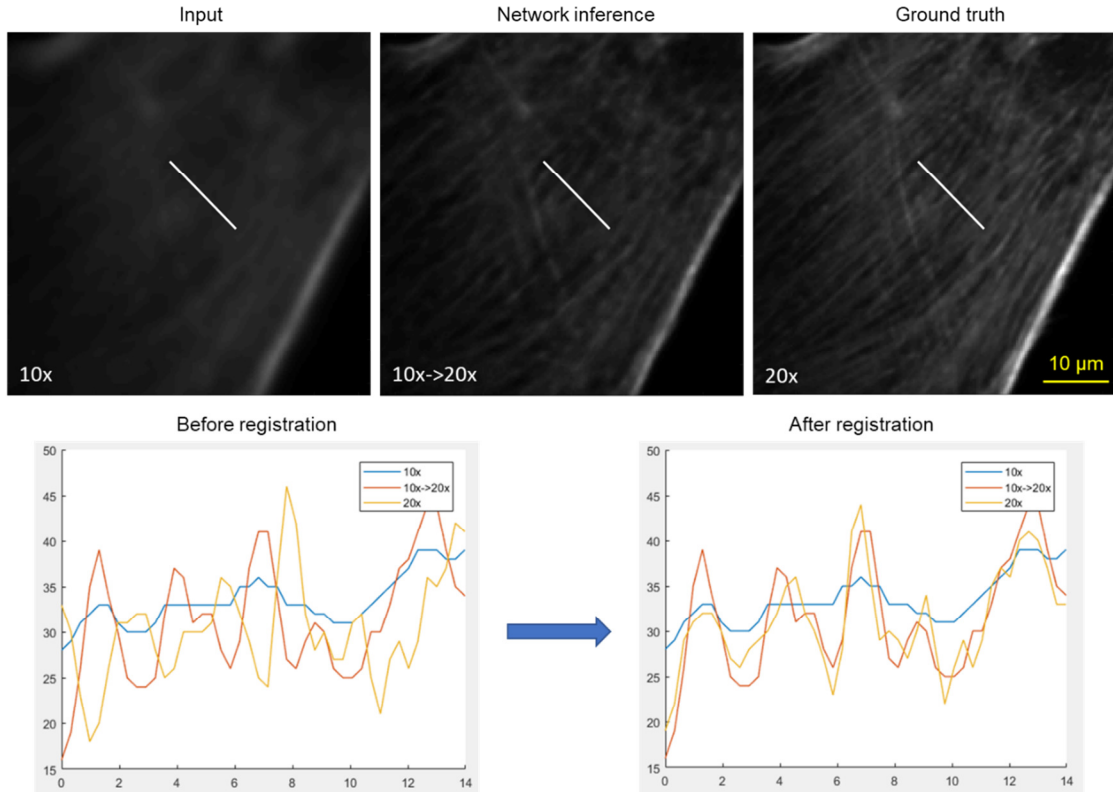
Important Notes

1. None of the test images were used in the training process.
2. The generalization folder contains blood vessel, mitochondria, and mouse kidney images, which can be super-resolved by the Wide-field/TxRed model trained with F-actin images only, as demonstrated in our manuscript.
3. **We emphasize that a “new” network model should be trained for achieving optimal super-resolution performance on input images corresponding to different types of samples or captured with a new experimental setup or microscope.**



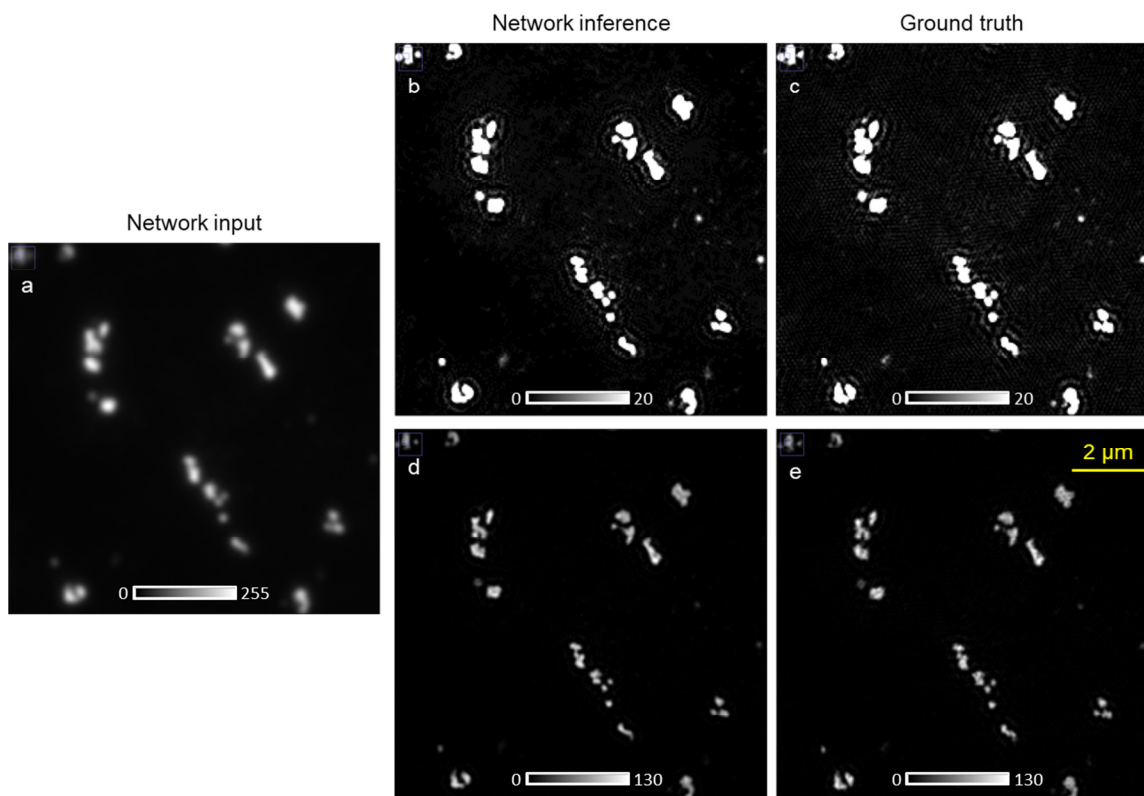
Example images: correct model's inference vs. another model's inference.

4. **Note that the test images included in this Plugin (i.e., lower resolution input images and the corresponding target label images) are not cross-registered to each other and therefore there might be random amounts of relative changes between the cross-sectional profiles that are taken from the network output and the label image.** This 2D cross-registration process was done in our training image set (see Supplementary Note 4 and Supplementary Figure S16) using small patches of images, since there is no uniform registration function across the whole field-of-view, as it changes from field-of-view to field-of-view due to aberrations etc.



Example images: Cross-sectional profiles drawn before and after the 2D image cross-registration. Blue is taken from the input image, yellow is taken from the ground truth image, the red is taken from the network output.

5. **Note that the aberrations or artifacts observed in some of the ground truth training images can couple back to the network's inference and result in some residual artifacts in the network output.** If the ground truth training image set is not dominated with such artifacts, the impact of this would be negligible, close to the noise floor of an image, as shown in the sample image below.



Example images: Artifacts observed in the ground truth training images can be coupled back to the network inference. Note the adjusted colormaps in (b,c) to better highlight these artifacts, while saturating some features. (d,e) show the same comparison between the network output and the ground truth, without any saturation.

Such residual artifacts can be further reduced by pre-selection of the training ground truth images to be free from such artifacts (if possible) or through an additional loss term applied to suppress such features during the training process.

6. In case the ground truth images of the same static sample exhibit image-to-image variations (such as the STED images reported in our manuscript, see Figure 5 of the main text and Figure S13 bottom row), one approach is to use the average of a few ground truth images for the network's inference comparison and/or the training process. In order to create a better ground truth image, before such an image averaging process image cross-registration among the ground truth fields-of-view could be needed depending on the relative drift of the imaging instrument, from image to image.